目 录

第一章 简介 2

1.1 硬件系统介绍及安装说明 2

1.2 软件系统介绍 2

1.3 产品特性 3

1.4 系统需求 3

第三 章 SDK3000 安装指南 4

第二章 SDK3000 函数说明 8

第四章 使用说明 23

在启动DEMO前必须使用REGSVR32注册Deinterlace.ax Filter文件(Deinterlace.ax Filter文件必须和MIX.DLL在同一个目录方可注册成功),否则会提示启动设备失败。

如果你还有其它方面的问题,请到深圳天敏视讯科技网址 http://www.10Moons.com

版本 1.0.0.0 发行日期: 2004.10

深圳天敏视讯科技版权所有

关于此手册

以下为本手册的内容介绍：

本开发包说明书包括以下几个方面:

第一章　简介

1.1　硬件系统介绍及安装说明

1.2　软件系统介绍

1.3　软件功能介绍

第二章　安装指南

第三章　函数说明

第四章　使用说明

第一章 简 介

本文档提供给使用10MOONS SDK3000系统产品做开发的用户使用.该系统要求硬件配置CPU PIII1.0G,内存128MB,硬盘40G以上,软件运行在 Windows 2000 、Windows XP，支持DirectX Show。

本文档只对系统开发中用到的动态库导出函数及使用方法做详细说明.您可以用VC++或BC++等开发工具做开发,使用时请包含DLL和LIB文件. SAA7130为10MOONS SDK3000产品中的视频主芯片，所有的视频处理都由它来完成。

天敏SDK3000卡是一款专门针对系统开发商及电脑DIY发烧友的高品质PCI 视频卡。它具有高品质的视频采集性能,具备高速PCI总线,兼容即插即用(PNP),支持一机多卡。我们给您提供功能全面的二次开发包(以下简称SDK)。您可以选择VisualBasic、VisualC++、Delphi等多种编程语言通过SDK进行开发，SDK中包含DLL动态库及其详细说明。您可通过SDK控制图像的输入端口,图像亮度,对比度,色度,灰度等输入信号,动态截取图像,以AVI格式进行录像,侦测图像是否有移动目标等等。我们向您提供完善的二次开发技术支持,并可根据您的要求度身定做SDK。

他是市场上我公司率先先推出的采用Philips 7130芯片的视频卡。Philips 7130芯片是一颗9bit ADC,相对于8bit ADC BT878芯片来说不管是图像质量还是颜色的饱和度方面都要强很多。我公司针对市场的需求,在开发过程中增加了一些新功能以满足不同客户的要求。(如可采集单场,单帧,连续帧,间隔几帧,连续相邻帧的图象,精确到帧等)

1.1 硬件系统介绍

PCI 卡为单卡 1 块 SAA7130 芯片，可以传输 4 路 Video 和 1 路 SVideo 实时视频信号。同时，在机器性能允许的情况下可实现单机多卡运行。SDK3000 支持多制式的视频源（PAL 制、NTFS 制、SECAM 制）。使用 1 块卡输出的帧率，视频源 50HZ 输出的帧率最大为 25 帧，视频源 60HZ 输出的帧率最大为 30 帧。

1.2 软件系统介绍

本套软件实现硬件功能有：实时 Preview 显示，实时提供多种数据压缩格式。YUY2 实现图像的快速抓拍，允许用户存为 BMP，JPG 格式的文件，提供多种视频源的原始视频（YUY2,RGB32， RGB24,RGB555）。

该软件包括驱动文件,SDK文件,APP文件三部分。

1.3 产品特性

1、频预览：提供 Preview 方式的视频预览。

2、视频捕获：为用户提供 YUY2 格式枚举压缩和设置压缩方式。

3、设置帧率大小：允许用户设置帧率大小来进行预览或捕获。

4、允许用户设置视频输出的尺寸。

5、提供动态检测方式。

6、提供 YUY2 视频格式的抓图，图像存为 JPEG 格式文件或 BMP 格式的文件。

7、支持 4 路 Video 和 1 路 SVideo 视频源，允许用户进行视频源之间的切换。

8、允许用户设置视频源属性：亮度、对比度、饱和度、色度。

9、IO 报警：提供 IO 输出，IO 输入。

10、视频预览映像：提供预览视频的任意缩放和全屏方式。

11、设置文本叠加： 允许用户设置文本叠加，和时间叠加。

12、显示属性页对话框：为用户提供显示页对话框交互方式。

13、视频源设置制式。

14、为用户提供 YUY2,RGB32,RGB24,RGB555 视频源回调方式

1.4 系统需求

CPU PIII1.0G；内存 128MB；硬盘 40G 以上

Windows 2000 、Windows XP

支持 DirectX Show

第二章　　安装指南

3.1 驱动程序安装(以windows xp为例做说明)

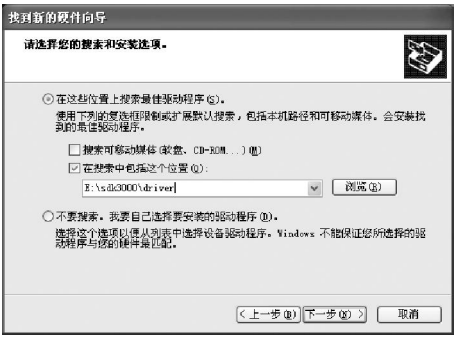
驱动程序的安装以windows xp为例做说明,并不是表明该驱动程序不可以在别的操作系统中使用，SDK3000驱动程序支持windows 2000。

安装前注意：可能在某些操作系统中安装时没有提示找到新硬件。

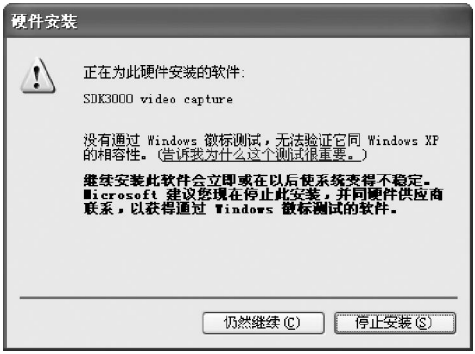
你可以在设备管理器中找到如图中的设备，请您删除它们，再重新启动系统。

由于SDK3000卡具有即插即用的功能，所以在硬件安装完毕后，启动wiodows时会自动检测到SDK3000卡，此时系统将提示“发现新硬件”并弹出：“找到新硬件向导”。

1） 请选择“从列表或指定位置安装（高级）”，然后单击“下一步”。



2） 插入随卡附送的天敏SDK3000驱动光碟，选择“在这些位置上搜索最佳驱动程序”，单击“浏览”，选择SDK300驱动所在目录，单击“下一步”。



3） windows xp 提示您，它无法了解此驱动程序没有通过Microsoft 测试，请单击“仍然继续”。

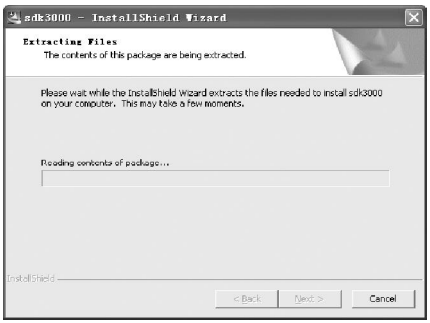
4）找到新硬件向导正在拷贝文件。

6） 单击“完成”，向导已经帮助您安装好SDK3000 的设备。

7） 请确认您的系统设备管理器是否包含SDK3000设备，并且没有任何感叹号（！）或是“X”号。

3.2 应用软件的安装(以windows xp为例做说明)

应用软件的安装以windows xp为例做说明,并不是表明该驱动程序不可以在别的操作系统中使用，SDK3000应用软件支持windows 2000。



1） 将天敏产品附送的光碟放入CDO-ROM驱动器中并运行：x:\sdk3000\app\sdk3000.exe 进入SDK3000的安装界面。

2） 选择安装语言，选好后单击“确定”。

3） 单击“ 下一步 ”。

4）选择目标文件夹，可以使用默认的目标文件夹：C：\Program Files\10moons\sdk3000 也可以点击旁边的“ 浏览 ”按钮选择目标文件夹，选好后点击“ 下一步 ”。

6）单击“ 完成 ”按钮，Windows 已经完成了 SDK3000 的应用软件安装。

第三章 10MOONS SDK3000 函数说明

sdk3000_7130.DLL 使用说明 (for VC++ 6.0)

操作系统：Windows 2000/winxp

一、编译环境

1、 将SDK安装目录下的Inc 和 Lib 目录路径分别加入VC++工程设置的 “C/C++ - Preprocressor – Additional include directories”和“Link – Input – Additional library path”中。

2、 将 Lib\sdk3000_7130.LIB 文件加入工程设置的“Link – General – Object/library module”中 。

3、 将 Inc\sdk3000_7130.h 文件加入工程

4、 将 Exe\sdk3000_7130.DLL文件拷入系统目录或其他系统能找到的目录中。

5、 开始编写代码。

该SDK文件即sdk3000_7130.DLL,输出功能在头文件sdk3000_7130.H文件中定义

二、函数介绍：

3.1 数据常量介绍

static const ULONG BrightnessDefault = 0x80;

说明：亮度缺省值

static const ULONG ContrastDefault = 0x44;

说明：对比度缺省值

static const ULONG SaturationDefault = 0x40;

说明：饱和度缺省值

static const ULONG HueDefault = 0x00;

说明：色度缺省值

static const ULONG SharpnessDefault = 0x00;

说明：清晰度缺省值

#define MIN_VAMP_BRIGHTNESS_UNITS 0

说明：亮度最小值

#define MAX_VAMP_BRIGHTNESS_UNITS 255

说明：亮度最大值

#define MIN_VAMP_CONTRAST_UNITS -128

说明：对比度最小值

#define MAX_VAMP_CONTRAST_UNITS 127

说明：对比度最大值

#define MIN_VAMP_HUE_UNITS -128

说明：色度最小值

#define MAX_VAMP_HUE_UNITS 127

说明：色度最大值

#define MIN_VAMP_SATURATION_UNITS -128

说明：饱和度最小值

#define MAX_VAMP_SATURATION_UNITS 127

说明：饱和度最小值

#define MIN_VAMP_SHARPNESS_UNITS -8

说明：清晰度最小值

#define MAX_VAMP_SHARPNESS_UNITS 7

说明：清晰度最小值

3.2 数据结构介绍:

```
typedef enum
{
IMAGE_BMP = 1, //BMP文件格式
IMAGE_JPG = 2 //JPG文件格式
}
enImageType;
```

说明：图像格式.

```
typedef enum
{
VIDEO_IN_1 = 0, //AV视频源1
VIDEO_IN_2 = 1, //AV视频源2
VIDEO_IN_3 = 2, //AV视频源3
VIDEO_IN_4 = 3, //AV视频源4
SVIDEO_IN = 4 //SVIDEO视频源
```

```
}VIDEOSOURCE;
```
说明：输入视频源端口

```
typedef enum tagVideoProcAmpProperty
{
VideoProcAmp_Brightness = 0,
VideoProcAmp_Contrast = VideoProcAmp_Brightness + 1,
VideoProcAmp_Hue = VideoProcAmp_Contrast + 1,
VideoProcAmp_Saturation = VideoProcAmp_Hue + 1,
VideoProcAmp_Sharpness = VideoProcAmp_Saturation + 1,
VideoProcAmp_Gamma = VideoProcAmp_Sharpness + 1,
VideoProcAmp_ColorEnable = VideoProcAmp_Gamma + 1,
VideoProcAmp_WhiteBalance = VideoProcAmp_ColorEnable + 1,
VideoProcAmp_BacklightCompensation = VideoProcAmp_WhiteBalance + 1,
VideoProcAmp_Gain = VideoProcAmp_BacklightCompensation + 1
}VideoProcAmpProperty;
```
说明：视频处理属性.

```
typedef enum
{
VideoProperty_Brightness = VideoProcAmp_Brightness, // 亮度
VideoProperty_Contrast = VideoProcAmp_Contrast, // 对比度
VideoProperty_Hue = VideoProcAmp_Hue, // 色度
VideoProperty_Saturation = VideoProcAmp_Saturation, // 饱和度
VideoProperty_Gamma = VideoProcAmp_Gamma, //Gamma 校验
VideoProperty_ColorEnable = VideoProcAmp_ColorEnable,
VideoProperty_WhiteBalance= VideoProcAmp_WhiteBalance, // 白平衡
VideoProperty_BacklightCompensation = VideoProcAmp_BacklightCompensation,
VideoProperty_Sharpness = VideoProcAmp_Sharpness,
VideoProperty_Gain = VideoProcAmp_Gain,
}
VideoProperty;
```
说明：视频属性.

```
typedef enum
```

```
{
PropertyDlg_VideoCaptureFilter = 0,//SDK3000 视频设备属性页
PropertyDlg_VideoCapturePin, //SDK3000 视频设备输出引脚属性页
PropertyDlg_VideoCrossbar, //Video Crossbar 属性页
PropertyDlg_DeinterlaceFilter, //Deinterlace 设备属性页
PropertyDlg_DeinterlaceCapturePin, //Deinterlace 设备捕获引脚属性页
PropertyDlg_DeinterlacePreviewPin //Deinterlace 设备预览引脚属性页
}
PropertyDialog;
```

说明：SDK3000 属性页.

```
typedef enum{
STATE_STOP = 0,
STATE_RUN,
STATE_PAUSE
}
GRAPH_STATE;
```

说明：视频运行状态.

```
typedef enum
{
VideoSubType_None = -1,
VideoSubType_RGB555 = 0,
VideoSubType_RGB24,
VideoSubType_YUY2,
VideoSubType_RGB8,
VideoSubType_RGB32
}
VideoSubType;
```

说明：视频格式。VideoSubType_YUY2 格式支持 SDK3000 所有功能。视频制式除 VideoSubType_YUY2 外，只能支持视频原始回调方式。

```
typedef enum
{
VideoStandard_None = 0x00000000,
```

```
VideoStandard_NTSC_M = 0x00000001,
VideoStandard_NTSC_M_J = 0x00000002,
VideoStandard_NTSC_433 = 0x00000004,
VideoStandard_PAL_B = 0x00000010,
VideoStandard_PAL_D = 0x00000020,
VideoStandard_PAL_H = 0x00000080,
VideoStandard_PAL_I = 0x00000100,
VideoStandard_PAL_M = 0x00000200,
VideoStandard_PAL_N = 0x00000400,
VideoStandard_PAL_60 = 0x00000800,
VideoStandard_SECAM_B = 0x00001000,
VideoStandard_SECAM_D = 0x00002000,
VideoStandard_SECAM_G = 0x00004000,
VideoStandard_SECAM_H = 0x00008000,
VideoStandard_SECAM_K = 0x00010000,
VideoStandard_SECAM_K1 = 0x00020000,
VideoStandard_SECAM_L = 0x00040000,
VideoStandard_SECAM_L1 = 0x00080000,
}
VideoStandard;
```

说明：视频制式.

```
typedef enum
{
Even_Field = 0, //偶场
Odd_Field, //奇场
Odd_Even_Field, //奇场、偶场
Interlaced_Field //复合场
}
VideoFieldType;
```

说明：场类型.

```
typedef struct
{
LONG cfHeight; //字符高度
```

```
LONG cfWidth; //字符宽度
LONG cfWeight; //字符粗细
LONG cfSpacing; //字符间距
BOOL cfItalic; //斜体
BOOL cfUnderline; //下划线
BOOL cfStrikeOut; //删除线
COLORREF cfBkColor; //背景色
COLORREF cfTextColor; //前景色
TCHAR cfFaceName[32]; //字体名
}
OSDPARAM;
```

说明：视频叠加参数.

```
typedef struct
{
long nValue; //当前值
long nMin; //最小值
long nMax; //最大值
long nStepDelta; //最小步进值
long nDefault; //默认值
long nCapsFlags; //风格
}
VIDEOPROPERTYRANGE;
```

说明：视频属性范围.

```
typedef struct
{
VideoSubType subtype; //视频格式
RECT rcSource; //视频源范围
RECT rcTarget; //目标视频范围
DWORD dwBitRate; //比特率
DWORD dwBitErrorRate; //比特错误率
LONGLONG AvgTimePerFrame; //帧率
BITMAPINFOHEADER bmiHeader; //宽、高、颜色位率等
}
```

VIDEOSTREAMINFO;

说明：视频流信息.

3.3 SDK 导出函数说明：

typedef void (CALLBACK *PrcCbMotionDetect)(long lnCardID, BOOL bMove, LPVOID lpContext);

说明： 视频动态检测回调.

参数： lnCardID： 返回的卡号.

bMove： TRUE 表示运动，FALSE 表示停止.

lpContext： 文本内容.

返回值：无

typedef void (CALLBACK *PrcCapSourceStream)(long lnCardID, long pBuf, long lnWidth, long lnHeight, long lnBiCount);

说明: 捕获原始流回调.

参数： lnCardID： 返回的卡号.

pBuf： 缓冲区.

lnWidth： 视频宽度.

lnHeight： 视频高度.

lnBiCount： 每一象素所需字节数.

返回值：无

typedef void (CALLBACK *PrcIoAlertCallBack)(long lnCardID);

说明： IO报警回调.

参数： lnCardID： 返回的卡号.

返回值：无

typedef void (CALLBACK *PrcPicMessage)(long lnCardID, long pBuf, long lnWidth, long lnHeight);

说明：抓图回调.

参数：lnCardID：返回的卡号.

pBuf：缓冲区.

lnWidth：视频宽度.

lnHeight：视频高度.

返回值：无

备注：回调的图像数据格式为 RGB24,每 1 象素占三个字节。

HRESULT WINAPI VCAInitSdk();

说明： 初始化 SDK.

参数： 无。

返回值：S_OK:成功.

S_FALSE:失败.

void WINAPI VCAUnInitSdk()

说明： 释放 SDK 资源.

参数： 无

返回值：无

HRESULT WINAPI VCAGetDevNum(int *pDevNum)

说明： 得到SDK3000设备总数.

参数： pDevNum： 设备总数指针

返回值：S_OK:成功.

S_FALSE:失败..

HRESULT WINAPI VCAConnectDevice(int nCards,

BOOL bOverlay,

HWND hParentWnd,

CSize szVidSize,

VIDEOSOUR enVideoSource,

int nFrameRate = 25,

VideoSubType enVideoSubType = VideoSubType_YUY2)

说明： 连接设备.

参数： nCards： 卡号.

bOverlay： 是否使用 OVERLAY 表面

hParentWnd：显示窗口句柄

szVidSize： 窗口尺寸

enVideoSource：视频端口

nFrameRate：帧率

enVideoSubType：视频格式

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCADisConnectDevice(int nCards)

说明： 断开设备.

参数： dwCard：卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetCaptureFile(int nCards, LPCTSTR szFileName);

说明： 设置捕获文件名.

参数： nCards： 卡号.

szFileName：文件名

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAStartCapture(int nCards);

说明： 开始捕获.

参数： nCards： 卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAStopCapture(int nCards)

说明： 停止捕获.

参数： nCards：卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAIsCapturing(int nCards, BOOL* bIsCapturing)

说明：是否正在捕获.

参数： nCards：卡号.

bIsCapturing：是否正处于捕获状态

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetWindowPos(int nCards, RECT rc)

说明： 设置视频在显示窗口中的区域.

参数： nCards：卡号.

rc：视频在显示窗口中的区域

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAEnableMotionDetect(int nCards, BOOL bEnaDetect, BYTE *pAreaMap, LONG nSizeOfMap, LONG nPersistTime, LPVOID lpContext, PrcCbMotionDetect OnObjectMove)

说明： 设置动态检测.

参数： nCards：卡号.

bEnaDetect： TRUE：允许动态检测，后续参数有效。

FALSE：禁止动态检测，系统忽略后续参数。

pAreaMap： 系统将捕获视频划分为16*16点阵的动态检测区域，按从左到右、从上到下的次序编号，依次为0、1、2、…，其中每一编号分别对应pAreaMap中的一个BYTE，pAreaMap数组中某一单元的BYTE值表示对应子块的动态检测灵敏度，灵敏度范围从0-49，0表示最高灵敏度，49表示最低灵敏度，如该值大于49，系统不对该块进行动态检测。

nSizeOfMap： pAreaMap数组的大小，pAreaMap数组应由用户分配，其大

小计算公式如下：

设nCapWidth,nCapHeight分别为捕获视频的宽度和高度，nDetectWidth,nDetectHeight为以子块为单位的检测块宽度和高度,pAreaMap数组的大小定义为nDetectBlocks。

nDetectWidth = (long)(nCapWidth / 16);

nDetectHeight = (long)(nCapHeight / 16);

nDetectBlocks = nDetectWidht * nDetectHeight;

即：系统以16为模对捕获视频宽度和高度进行划分，多余

部分则自动丢弃而不做检测。

nPersistTime： 用户设定的静止检测持续时间(以秒为单位)，即当系统检测到物体的一次运动后，当持续检测到nPersistTime 秒没有发生物体运动才认为物体进入静止状态，该值取值范围从0 - 30s。

lpContext： 它会作为回调函数的最后一个参数返回给用户，用户可以在

该成员中保存一些私有信息。如果不需要的话，可以传入

NULL值。

OnObjectMove：用于通知用户动态检测状态的回调函数指针。每当系统检测到有物体移动时，该函数被调用一次。当物体停止移动时，该函数再被调用一次，不能为NULL值。

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCACapturePicture(int nCards, LPCTSTR szFileName, enImageType enImgType, RECT* rcImg = NULL, long nQuality = 100)

说明： 抓图.

参数： nCards：卡号

szFileName：文件名

enImgType：图像格式

rcImg：在显示窗口中抓图的区域，NULL 表示整个窗口。

nQuality：抓图质量。范围为 1～100，当图像为 IMAGE_JPG 时有效。

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASwitchVideoSource(int nCards, VIDEOSOURCE enVideoSource)

说明： 切换视频源

参数： nCards：卡号

enVideoSource：视频端口

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetVideoPropertyValue(int nCards, VideoProperty eVideoProperty, long value)

说明： 设置视频属性.

参数： nCards：卡号.

eVideoProperty：视频属性

value：设置值

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAGetVideoPropertyValue(int nCards, VideoProperty eVideoProperty, VIDEOPROPERTYRANGE* pVPR)

说明： 得到视频属性.

参数： nCards：卡号.

eVideoProperty：视频属性

pVPR：返回的视频属性

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAIOAlertOut(int nCards, BOOL bHighLevel)

说明： IO 输出.

参数： nCards：卡号.

bHighLevel：是否高电平

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAInitIOAlertIn(int nCards, PrcIoAlertCallBack ppCall)

说明： 初始化 IO 输入.

参数： nCards： 卡号.

ppCall：回调函数的指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAStartIOAlertIn(int nCards, BOOL bHigh)

说明： 开始等待 IO 输入.

参数： nCards：卡号.

bHigh：是否高电平出发，为 TRUE 表示高电平触发，为 FALSE 表示低电平触发.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAStopIOAlertIn(int nCards)

说明： 停止等待 IO 输入.

参数： nCards：卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetTextOSD(int nCards, BOOL bEnableOSD, TCHAR* pText, POINT ptTopLeft, BOOL bTransparent, OSDPARAM* pOSD, HWND nWndMain)

说明： 设置文本叠加.

参数： dwCard：卡号.

bEnableOSD：是否允许叠加.

pText：叠加文本.

ptTopLeft：在窗口区域叠加位置文本的位置.

bTransparent：是否透明.

pOSD：叠加参数.

nWndMain：主窗口句柄.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetDateTimeOSD(int nCards, BOOL bEnableOSD, POINT ptTopLeft, BOOL bTransparent, OSDPARAM* pOSD, HWND nWndMain)

说明： 设置时间叠加.

参数： dwCard：卡号.

bEnableOSD：是否允许叠加.

ptTopLeft：在窗口区域叠加位置文本的位置.

bTransparent：是否透明.

pOSD：叠加参数.

nWndMain：主窗口句柄.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCADisplayPropertyDialogs(int nCards, PropertyDialog id, HWND hParentWnd, LPCTSTR szCaption)

说明： 显示属性页对话框.

参数： nCards：卡号.

id：属性对话框 ID 号.

hParentWnd：父窗口句柄.

szCaption：对话框字符串

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCARun(int nCards)

说明：运行视频流.

参数： nCards：卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAPause(int nCards)

说明： 暂停视频流.

参数： nCards： 卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAStop(int nCards)

说明： 停止视频流.

参数： nCards： 卡号.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAGetStreamStatus(int nCards, GRAPH_STATE* pStatus)

说明: 得到视频流状况.

参数： nCards：卡号.

pStatus：视频流状态指针

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetFrameRate(int nCards, int nFrameRate)

说明： 设置视频帧率.

参数： nCards：卡号.

nFrameRate: 帧率.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAGetVideoInfo(int nCards, VIDEOSTREAMINFO * pVSI)

说明： 得到视频信息.

参数： nCards：卡号.

pVSI：视频流信息指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetVideoInfo(int nCards, VIDEOSTREAMINFO * pVSI)

说明： 设置视频信息.

参数： nCards：卡号.

pVSI：视频流信息指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetFullScreen(int nCards, BOOL bFullSrc)

说明： 设置视频大小为全屏.

参数： nCards：卡号.

bFullSrc：是否以全屏方式显示.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAIsFullSrcMode(int nCards, BOOL* pFullSrc)

说明： 是否视频为全屏模式.

参数： nCards：卡号.

pFullSrc：是否以全屏方式显示的指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAGetVideoStandard(int nCards, VideoStandard* pVideoStandard, long* pAvailable)

说明： 得到视频制式.

参数： nCards：卡号.

pVideoStandard：当前的视频制式指针.

pAvailable：硬件支持的所有制式指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetVideoStandard(int nCards, VideoStandard vs)

说明： 设置视频制式.

参数： nCards：卡号.

vs：视频制式.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAEnumVideoCompressor(int nCards, char* arrCompDevName)

说明： 枚举视频压缩格式.

参数： nCards：卡号.

arrCompDevName：视频压缩设备指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAGetCurrentVideoCompressor(int nCards, LPSTR pCompDevName)

说明： 得到视频当前压缩格式.

参数： nCards：卡号.

pCompDevName：视频当前压缩格式指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCASetCurrentVideoCompressor(int nCards, LPCTSTR pCompDevName)

说明： 枚举视频当前压缩格式.

参数： nCards：卡号.

pCompDevName：视频当前压缩格式指针.

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAEnableCapSourceStream(int nCards, BOOL bCapStream, VideoFieldType enVideoFieldType, PrcCapSourceStream pCapStream)

说明： 设置视频原始回调格式.

参数： nCards：卡号.

bCapStream：是否允许回调.

enVideoFieldType：场类型

pCapStream：原始回调函数指针

返回值：S_OK:成功.

S_FALSE:失败.

HRESULT WINAPI VCAEnablePicMessage(int nCards, BOOL bPicMessage, PrcPicMessage pPicMessage)

说明： 设置抓图回调.

参数： nCards：卡号.

bPicMessage：是否允许回调.

pPicMessage：原始回调函数指针

返回值：S_OK:成功.

S_FALSE:失败.

第四章　10MOONS SDK3000 安装说明

安装使用说明:

系统安装完成后会在安装路径下生成如下目录

安装路径：

|----------CAP7134.SYS（SAA7134 驱动）

|----------sdk3000_7130.dll(SAA7134 二次开发所需的动态库)

|----------sdk3000_7130.h(SAA7134 二次开发所需的头文件)

|----------sdk3000_7130.lib(SAA7134 二次开发所需的 lib 库)

|----------Mpeg4Codec(XVID Mpeg4 压缩引擎)

|----------Mix.dll(字幕叠加所需的动态库)

|----------SAA7134Demo.exe(Demo 程序可执行文件)

|---------- SAA7134Demo(Demo 源程序代码)

注：如果您想使用本软件的 MPEG4 压缩功能，请先注册 XVID 压缩引擎,安装方法为：鼠标右键单击 Mpeg4Codec 目录下的 Xvid.inf 文件，在弹出式菜单中选择“安装”即可。

另：其他程序不需注册，直接使用即可。